(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 April 2002 (25.04.2002)

PCT

(10) International Publication Number
WO 02/33590 A1

(51) International Patent Classification⁷: G06F 17/30, G06N 5/02

(21) International Application Number: PCT/SG01/00217

(22) International Filing Date: 18 October 2001 (18.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/690,738 18 October 2000 (18.10.2000) US

(71) Applicant *(for all designated States except US)*: **KENT RIDGE DIGITAL LABS** [SG/SG]; 21 Heng Mui Keng Terrace, Singapore 119613 (SG).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **NG, SEE, Kiong** [SG/SG]; Blk 7 Haig Road, #10-439, Singapore 430007 (SG). **WONG, Lim, Soon** [MY/MY]; 19 Jalan 11 Taman Melawati, 53100 Kuala Lumpur (MY).

(74) Agent: **AXIS INTELLECTUAL CAPITAL PTE LTD.**; 19B Duxton Hill, Singapore 089602 (SG).

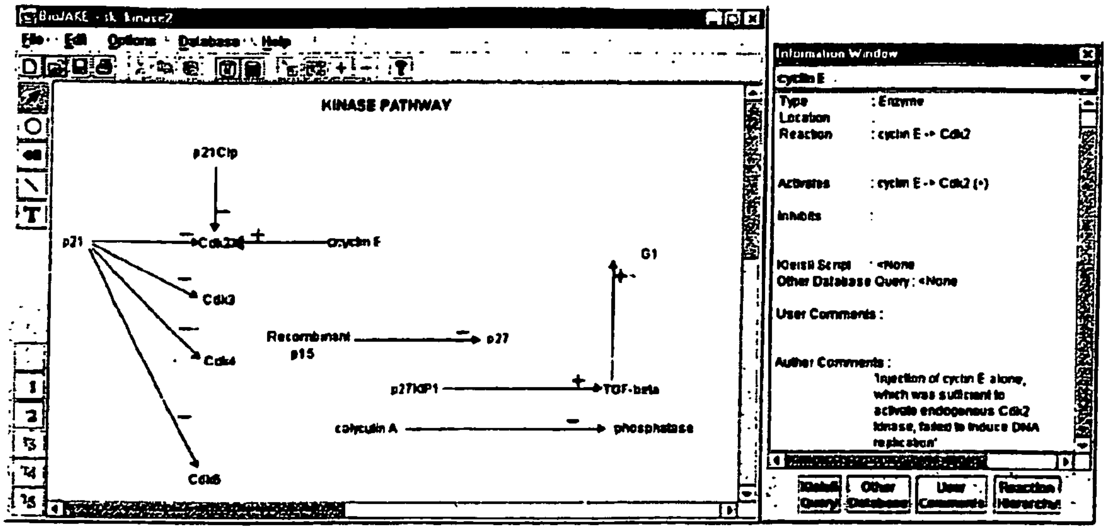
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

[Continued on next page]

(54) Title: A PROTEIN INTERACTION EXTRACTION SYSTEM



Type : Enzyme
Location :
Reaction : cyclin E -> Cdk2

Activates : cyclin E -> Cdk2 (+)

Inhibits :

Kleisli Script : <None
Other Database Query : <None

User Comments :

Author Comments :
'Injection of cyclin E alone, which was sufficient to activate endogenous Cdk2 kinase, failed to induce DNA replication'

(57) Abstract: The Protein Interaction Extraction System (PIES) is a means for automatic pathway discovery from online text abstracts. It combine technologies that (a) retrieve abstracts from online sources, (b) extract relevant protein interaction information from free texts, (c) present the extracted information graphically and intuitively, and optionally (d) allow (possibly customised) queries to be attached and launched from the graphical interface. It can further support sophisticated manipulations of the extracted pathways. It can also be set to periodically/routinely scan online scientific literature for automatic discovery of new protein interactions, giving modern scientists the necessary competitive edge in managing the information explosion in this electronic age.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A PROTEIN INTERACTION EXTRACTION SYSTEM

TECHNICAL FIELD OF THE INVENTION

The present invention relates to the field of research, particularly the searching, scanning and / or analysis of a voluminous amount of information available in databases where the latest scientific discoveries are often lodged and first reported online and are accessible by scientists worldwide. More particularly, the present invention relates to the research undertaken and reported related to the biotechnology and pharmaceutical industries.

BACKGROUND OF THE INVENTION

A large part of the information required for biology research can only be found in free-text form, as in MEDLINE abstracts, or in comment fields of relevant reports, as in GenBank feature table annotations. Such information is important for many types of analysis, such as classification of proteins into functional groups, discovery of new functional relationships, maintaining information of material and methods, increasing the precision and relevance of hits returned by information retrieval systems, and so on. Such free-texts are increasingly available online; for example, MEDLINE abstracts have been accessible via the World Wide Web for many years now.

However, it is very challenging and time consuming for a scientist to sift through the accumulated abstracts and to monitor the newly published ones for the specific information he wants. An important example of the specific information is "what newly discovered proteins interact with a specific protein directly or indirectly via a small number of intermediaries". Imagine the frustrations of a scientist who has to read through hundreds (if not thousands) of abstracts to find out that "p21 inhibits Cdk2."

Prior art methods and apparatus are described in the following:

- K. Fukuda, *et al.,* "Toward information extraction: Identifying protein names from biological papers." *Proc. Pacific Symposium on Biocomputing,* 707-718, 1998.

- S. Goto, *et al.*, "Organizing and computing metabolic pathway data in terms of binary relations." *Proc. Pacific Symposium on Biocomputing,* 175-186, 1997.
- B. Jacq, *et al.*, "GIF-DB, a WWW database on gene interactions involved in Drosophila melanogaster development." *Nucleic Acids Research,* 25:67-71, 1997.
- M. Kanehisa, "A database for post-genome analysis." *Trends Genet.,* 13:375-376. 1997.
- P. Karp, *et al.*, "EcoCyc: Electronic Encyclopedia of E. coli Genes and Metabolism." *Nucleic Acids Research,* 26:50, 1998.
- Y. Ohta, *et al.*, "Automatic construction of knowledge base from biological papers." *Intelligent Systems for Molecular Biology*, 5:218-225, 1997.
- D. Proux, *et al.*, "Detecting gene symbols and names in biological texts: First step toward pertinent information extraction." *Genome, Informatics*, 9:72-80, 1998.
- W. Salamonsen, *et al.*, "BioJAKE: A tool for the creation, visualization and manipulation of metabolic pathways." *Proc. Pacific Symposium on Biocomputing*, 392-400, 1999.
- T. Sekimizu, *et al.*, "Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts." *Genome Informatics,* 9:62-71, 1998.
- T. Takai-Igarashi, T. Kaminuma, "A Pathway finding System for the Cell Signalling Networks Database." *In Silico Biology*, 1:0012, 1998.
- M. Tomita, *et al.*, "A virtual cell with 127 genes." *Proc. Ist Int'l Conference on Bioinformatics of Genome Regulation and Structure*, 97-103, 1998.
- M. Tomita, *et al.*, "E-CELL Project overview: Towards integrative simulation of cellular process." *Genome Bioinformatics*, 242-243, 1998.
- L. Wong. "Some Medline queries powdered by Kleisli." *ACCESS*, 25:8-9,

1998.

The main problems in existing prior art include their inability to perform automatic extraction, their inability to let a user (biologist) specify the specific type of protein interaction/pathways to extract, their inability to let a user (biologist) modify the visual presentation of pathways, their inability to let a user (biologist) attach queries to the visual presentation of pathways, and / or their inability to let a user (biologist) specify a schedule for monitoring new additions/changes to his pathways. Specifically, by way of example:

- Existing systems (Goto *et al.*., 1997; Jacq *et al.*, 1997; Kanehisa, 1997; Karp *et al.*, 1998; Igarashi, Kaminuma, 1998; PUMA; Boehringer Mannheim Biochemical Pathways Chart; Tomita *et al.*, 1998) lack a relatively automated means of extracting protein interaction information from public databases.
- Existing systems (Goto, *et al.*, 1997; Kanehisa, 1997; Karp *et al.*, 1998; Tomita *et al.*, 1998) principally concentrate on specific topics, such as metabolic pathways or on non-human organisms (Karp *et al.*, 1998; Jacq *et al.*, 1997; PUMA) or CSNDB (Igarashi, Kaminuma, 1998) that concentrates on cellular signal transduction of human. Thus far, it is not possible for an individual biologist to specify what kind of protein interactions he wishes to curate or monitor.
- Existing systems (Goto *et al.*, 1997; Jacq *et al.*, 1997; Kanehisa, 1997; Igarashi, Kaminuma , 1998; PUMA; Boehringer Mannheim Biochemical Pathways Chart; Tomita *et al.*, 1998) present a relatively static unalterable visualisation. Even the simulations are considered to produce static looking graphs which may not be suitable for visualising simultaneous activities of various genes involved in a pathway. Furthermore, the graphical presentation of pathways cannot be altered easily and queries cannot be easily attached to the graphical presentation.
- Existing systems (Goto *et al.*, 1997; Jacq *et al.*, 1997; Kanehisa, 1997; Karp *et al.*, 1998; Igarashi, Kaminuma, 1998; PUMA; Boehringer Mannheim

Biochemical Pathways Chart; Tomita *et al.*, 1998) do not annotate the protein interaction links in their pathway maps with the research articles upon which the knowledge was discovered. Verification is an important step with newly discovered non-textbook pathway maps. As the pathway maps in these systems are not hyperlinked to the World Wide Web, it is difficult for the scientist to verify and expand the pathway maps with the information available on the Internet.

- Existing systems (Salamonsen *et al.*, 1999) do not allow the graphical representation to be edited in sophisticated ways. For example, in order to merge two nodes and their associated information in such systems, the user has to carry out the following tedious steps: create a new node, copy information from the two nodes to be merged into this new node, copy all the links to the two nodes to this new nodes, delete all the links to the two nodes, and finally delete the two nodes. For example, these systems also provide no automated means for a user to merge two graphical representations or for a user to extend a graphical representation by automatically extracting additional information using additional search terms.

There are a number of problems associated with the prior art, and an object of the present invention is to provide a research system and method, which addresses prior art problems.

SUMMARY OF THE INVENTION

The present invention seeks to address at least one of the problems mentioned above. One aspect of the present invention may be referred to as 'PIES', meaning Protein Interaction Extraction System, and is directed to providing a means for automatic discovery and presentation of biological pathway from on-line text abstracts. The present invention is also adapted to (a) perform automatic extraction of protein interaction and other information from online scientific literature, (b) let a user/biologist specify the specific type of protein interaction/pathways to extract, (c) generate pathway maps hyperlinked to supporting research articles on the World Wide Web ; (d) let a user/biologist modify the visual presentation of pathways, (e) let a user/biologist attach queries to the visual presentation of pathways, and / or (f) let a user/biologist specify a schedule for monitoring new additions/changes to his pathways.

The PIES combines technologies that (1) retrieve research abstracts from online sources, (2) extract relevant information from the free texts, (3) present the extracted information graphically and intuitively, and optionally (4) allow (possibly customised) queries to be attached and launched from the graphical interface. It can also be set to periodically/routinely scan online scientific literature for automatic discovery of knowledge, giving modern scientists the necessary competitive edge in managing the information explosion in this electronic age.

Another aspect of the present invention is directed to a method and apparatus adapted to:

a. scanning information related to a selected topic,

b. extracting information based on predetermined criteria,

c. generating an initial graphical representation of the extracted information, the improvement comprising

d. Providing an updated graphical representation by reiterating a, b and c on new information, not yet scanned, and

e. alerting a user to the presence of new extracted information based on the new information.

Preferably, the extracted information is obtained using natural language processing as published in SK Ng, M. Wong, "Toward routine automatic pathway discovery from on-line scientific text abstracts", Genome Informatics, 10:104—112, December 1999.

Preferably, the extracted information is obtained using a keyword based search.

Preferably, updated representation is represented in a visually distinctive manner compared to the initial graphical representation. The manner in which the visual distinctiveness is rendered is not important, merely that is it visually distinctive. For example, a different colour may be used to show each update.

Another aspect of the present invention is directed to a more sophisticated method and apparatus to extend and manipulate the extracted protein interaction pathways by

a. allowing the user (biologist) to specify additional keywords and then carrying out the following tasks automatically: search the Internet using these additional keywords for more scientific text abstracts, extract additional protein interaction information from these abstracts, and extend the current pathways with these additional protein interaction information.

b. allowing the user (biologist) to specify two sets of extracted pathways and then automatically constructing a set of new pathways by merging these two sets of extracted pathways.

c. allowing the user (biologist) to specify one or more distinct nodes (which correspond to proteins) in a set of pathways and then automatically constructing a new set of pathways by merging these two nodes and their associated interactions and other information.

d. allowing the user (biologist) to specify one or more nodes in a set of pathways and then automatically constructing a new set of pathways by deleting the specified nodes and their associated interactions and other information.

e. allowing the user (biologist) to specify one or more nodes and a radius (in terms of number of intermediary nodes) and then automatically constructing a new set of pathways consisting of just the followings: the specified nodes, other nodes within the specified radius of any of the specified nodes, the interactions of these nodes, and their other associated information.

f. allowing the user (biologist) to specify two sets of pathways and automatically determining their differences and constructing a new set of pathways containing exactly their differences.

g. allowing the user (biologist) to specify a node and then automatically constructing a new set of pathways by replicating the specified node and its associated interactions and information.

h. allowing the user (biologist) to specify an edge and then automatically constructing a new set of pathways by deleting this edge.

i. allowing the user (biologist) to specify an edge and then automatically constructing a new set of pathways by reversing the direction of this edge.

j. allowing the user (biologist) to specify an edge and then automatically inverting the nature of this edge; in other words, make an inhibition edge into an activation edge and vice versa.

k. allowing the user (biologist) to specify a new edge and then automatically constructing a new set of pathways by adding this new edge.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the present invention will now be described by way of example with reference to the accompanying drawings, in which:

Figure 1 illustrates a keyword type search used for scanning.

Figure 2 illustrates an abstract of extracted information.

Figure 3 illustrates a graphical representation of the extracted information, in which the edge (from p21 to Cdk2) is highlighted in red to indicate that it is a newly discovered protein interaction.

DETAILED DESCRIPTION OF THE INVENTION

The basic idea of the PIES is outlined in the following steps:

The user describes the kind of online text abstracts he is interested in by filling up a form such as that shown in Figure 1. This specification (e.g. "universal kinase inhibitor") is recorded in a folder kept for that user.

The PIES then retrieves text abstracts satisfying his specification. This retrieval can be repeated automatically by the PIES at regular intervals, to monitor newly published abstracts. Some examples satisfying the specification (e.g. "universal kinase inhibitor") are given in Figure 2.

The PIES then identifies important sentences from these abstracts, such as:

> p21 effectively inhibits Cdk2, Cdk3, Cdk4, and Cdk6 kinases (Ki 0.5-15 nM) but is much less effective toward Cdc2/cyclin B (Ki approximately 400 nM) and Cdk5/p35 (Ki > 2 microM), and does not associate with Cdk7/cyclin H.

and extracts from them precise protein interactions such as:

p21 - - inhibit - -> Cdk2
p21 - - inhibit - -> Cdk3
p21 - - inhibit - -> Cdk4
p21 - - inhibit - ->Cdk6.

The PIES then creates a graphical presentation of these protein interactions. The arcs in the presentation can be hyperlinked to Medline articles from which that particular protein interaction information was extracted. The user can then be notified (by email or other means) and access the presentation stored in his folder. An example presentation is shown in Figure 3. If an old presentation already exists in the user's folder, the newly discovered protein interactions can be highlighted for the user, as shown in red in Figure 3. It is also possible for the user to edit and directly manipulate the graphical presentation. For example, the user points and clicks on two nodes and causes the two nodes and their associated interactions and information to be merged automatically. It is also possible for him to attach some standard or customised queries to the arcs and nodes in the presentation. An example standard queries is "retrieve the amino acid sequence corresponding to this protein."

Implementation details are now described.

ARCHITECTURE

The PIES is composed of five modules and an underlying logical representation of the extracted pathways. We first provide an overview of the purpose of each of these modules, depicted below.

```
user inputs domain selection     keywords
```

```
                 |
                \|/
  +-----------------------------+
  |   1.BioKleisli-Abstracts    |
  +-----------------------------+
                 |
                 | search Internet for abstracts
                \|/
  +-----------------------------+
  |         2.BioNLP            |
  +-----------------------------+
                 | extract interaction information
                \|/
  +-----------------------------+
  |     3.BioKleisli-Graphs     |
  +-----------------------------+
                 | manipulation of logical representation
                \|/
  +-----------------------------+
  |       4.Graph Layout        |
  +-----------------------------+
                 | layout coordinates
                \|/
  +-----------------------------+
  |         5.BioJAKE           |
  +-----------------------------+
                 |
                \|/
system outputs editable pathway maps
```

1. The BioKleisli-Abstracts module is a software module constructed using the Kleisli query system (SY Chung, L Wong, "Kleisli, a new tool for data integration in biology", Trends in Biotechnology, 17(9):351--355, 1999). It is our query engine for retrieving scientific abstracts from online bibliographic resources on the Internet.

2. The BioNLP module is a software module constructed to process the free texts in scientific abstracts retrieved by the BioKleisli-Abstracts module. It identifies protein names mentioned in the free texts and performs function word pattern matching to discover protein-protein interactions expressed in these abstracts (SK Ng, M Wong, "Toward routine automatic pathway discovery from on-line scientific text abstracts", Genome Informatics, 10:104--112, 1999).

3. The BioKleisli-Graphs module is a software module also constructed using the

Kleisli query system. It is our query engine that provides the logical representation of the extracted pathways. It also provides operations for sophisticated manipulations of the logical representation. These operations include:

- allowing the user (biologist) to specify additional keywords and then carrying out the following tasks automatically: search the Internet using these additional keywords for more scientific text abstracts, extract additional protein interaction information from these abstracts, and extend the current pathways with these additional protein interaction information.
- allowing the user (biologist) to specify two sets of extracted pathways and then automatically constructing a set of new pathways by merging these two sets of extracted pathways.
- allowing the user (biologist) to specify one or more distinct nodes (which correspond to proteins) in a set of pathways and then automatically constructing a new set of pathways by merging these two nodes and their associated interactions and other information.
- allowing the user (biologist) to specify one or more nodes in a set of pathways and then automatically constructing a new set of pathways by deleting the specified node and its associated interactions and other information.
- allowing the user (biologist) to specify one or more nodes and a radius (in terms of number of intermediary nodes) and then automatically constructing a new set of pathways consisting of just the followings: the specified nodes, other nodes within the specified radius of any of the specified nodes, the interactions of these nodes, and their other associated information.
- allowing the user (biologist) to specify two sets of pathways and automatically determining their differences and constructing a new set of pathways containing exactly their differences.
- allowing the user (biologist) to specify a node and then automatically constructing a new set of pathways by replicating the specified node and its

associated interactions and information.

- allowing the user (biologist) to specify an edge and then automatically constructing a new set of pathways by deleting this edge.
- allowing the user (biologist) to specify an edge and then automatically constructing a new set of pathways by reversing the direction of this edge.
- allowing the user (biologist) to specify an edge and then automatically inverting the nature of this edge; in other words, make an inhibition edge into an activation edge and vice versa.
- allowing the user (biologist) to specify a new edge and then automatically constructing a new set of pathways by adding this new edge.

4. The Graph-Layout module is a software module for computing a visually pleasant layout for displaying the logical representation. In particular, this module assigns preliminary x-y co-ordinates that are to be used in the graphical representation of the extracted pathways.
5. The BioJAKE module is a visualisation engine to graphically display the information in the logical representation and to manage the constructed pathway maps in an intuitive manner to the user (Salamonsen *et al.*, "BioJAKE: A tool for the creation, visualisation, and manipulation of metabolic pathways", Proc. Pacific Symposium on Biocomputing, 392--400, 1999).

NOTATIONS

Before we proceed to the embodiment of each of the modules mentioned above, we have to introduce some notations that we will be using. These notations correspond to the scripting language supported and directly executed by the Kleisli query system (P. Buneman *et al.*, "Comprehension syntax", ACM SIGMOD Record, 23(1):87-96, 1994; SY. Chung, L. Wong, "Kleisli, a new tool for data integration in biology", Trends in Biotechnology, 17(9):351--355, 1999). We use these notations because they are high-level and thus easy to understand, because they are unambiguous, and because they are directly executable by the Kleisli query system and can thus be considered an effective implementation.

There are two kinds of things we need to describe: type and program. A type is a description of a data structure, a database schema, or a data type. A program is a description of the data manipulation and transformation steps or procedure.

Types are specified according to the grammar below:

$\texttt{t ::= unit | bool | num | string}$
$\quad | \quad (\#l_1\text{: } t_1, \ldots, \#l_n\text{: } t_n) \;|\; <\#l_1\text{: } t_1, \ldots, \#l_n\text{: } t_n>$
$\quad | \quad \{t\} \;|\; \{|t|\} \;|\; [t]$

The types unit, ..., string are the usual base or atomic types. The type $(\#l_1\text{: } t_1, \ldots, \#l_n\text{: } t_n)$ is a record type having fields $l_1, \ldots, l_n$ and these fields have types $t_1, \ldots, t_n$ respectively. The type {t} is a set whose elements have type t. The type {|t|} is a bag whose elements have type t. The type [t] is a list whose elements have type t. The main differences between set, bag, and list types are that in a set, the order and multiplicity of elements are ignored; in a bag, the order is ignored but multiplicity is important; and in a list, both order and multiplicity are important (V. Tannen *et al.*, "Logical and computational aspects of programming with sets/bags/lists", LNCS 510: Proc. 18th ICALP, pages 60--75, 1991). The type $<\#l_1\text{: } t_1, \ldots, \#l_n\text{: } t_n>$ is a "variant" type; the value of the variant type can be either of type $t_1, \ldots$, or t_n, but the value is explicitly tagged with the corresponding field label $l_1, \ldots$, or l_n. A variant type is similar to the concept of tagged union (R. Hull et. al, "The Format model: A theory of database organisation", J. ACM, 31(3):518--537, 1984) or the union type of the C programming language with explicit user-created tags.

The main programming constructs are the followings:

$\backslash x \rightarrow E$

$\{E \mid \backslash x \leftarrow E_1, \backslash y == E_2, E_3\}$

$E.\#l$

$f(E)$

$(\#l_1\text{: } E_1, \ldots, \#l_n\text{: } E_n)$

$<\#l\text{: } E>$

$\texttt{case } E \texttt{ of } <\#l_1\text{: } \backslash x_1> \rightarrow E_1 \texttt{ or } \ldots \texttt{ or } <\#l_n\text{: } \backslash x_n> \rightarrow E_n$

$\texttt{primitive } f == E$

Let us use the "meta" notation E^x to mean the value of E depends on (or with respect to) the current value of x. Then, the construct \x→E defines a function that takes input x and produces output E^x. The construct {E | \x ← E_1, \y == E_2, E_3} defines a set such that $E^{x,y}$ is in it if and only if x is in E_1, y is E_2^x, and $E_3^{x,y}$ is true for this particular x and y. The construct E.#l means the value of the field l of E. The construct (#l_1: E_1, ..., #l_n: E_n) builds a record having fields l_1, ..., l_n having values E_1, ..., E_n respectively. If f is a function, the construct f(E) means the result of applying f to E. The construct <#l: E> builds a variant whose value is E explicitly tagged by the label l. The construct case E of <#l_1: $\backslash x_1$> → E_1 or . . . or <#l_n: $\backslash x_n$> → E_n checks to see if E is a l_1, ..., or l_n variant; if it is a variant <#l_i: E'>, then it assigns E' to x_i and return the value of $E_i^{x_i}$. The notation primitive f == E defines a function or macro f whose value is E.

Comparison operations such as equality test, logical connectives such as negative, string operations such as substring test, and speciality bioinformatics operations such as sequence alignment, hidden Markov models, and access to MEDLINE, are provided in the Kleisli query system.

THE BIOKLEISLI-ABSTRACTS MODULE

Given a set, SPEC, of domain selection keywords provided by the user (biologist), the BioKleisli-Abstracts module retrieves relevant abstracts from MEDLINE and organises them for subsequent analysis by the BioNLP module. It can be implemented on top of the Kleisli query system using the following program script.

```
writefile
  { x
  | \s ← SPEC,
    \u ← ml-get-uid-general (s),
    \x ← ml-get-abstract-by-uid (u) }
to "articles" using machineout;
```

It works as follows. For each keyword s provided in the set SPEC, the Kleisli operation ml-get-uid-general is used to obtain unique identifiers of MEDLINE abstracts that match it. Then for each unique identifier u, the Kleisli operation ml-get-abstract-by-uid is used to obtain the corresponding MEDLINE abstract. Finally, store each abstract x into a file called articles.

The file articles has the following schema or type.

```
{ (#muid: num, #authors: string, #address: string,
  #title: string, #abstract: string, #journal: string) }
```

It is thus a set of records. Each record has a muid field which stores the unique identifier of the article, an authors field which stores names of authors, an address field which stores addresses of authors, a title field which stores the title of the article, an abstract field which stores the abstract of the article, and a journal field which stores the journal issue in which the article was published.

Other information can also be captured and stored in this file.

THE BIONLP MODULE

The details of the BioNLP module is given in the paper SK Ng, M. Wong, "Toward routine automatic pathway discovery from on-line scientific text abstracts", Genome Informatics, 10:104--112, December 1999. We provide an outline here.

The BioNLP module is a rule-based system that performs simple natural language processing on the extracted scientific abstracts using pattern matching. There are two major tasks in extracting protein-protein interaction information from scientific abstracts:

- Protein name identification. Straightforward use of a dictionary of protein names is inadequate in this domain because new names are continuously being invented and quoted in medical and biological papers. The names of the new proteins must therefore be identified by linguistic means.
- Information extraction. Co-occurrence of protein names in an article abstract, a sentence, or a phrase generally implies that the proteins are related in some way. Such co-occurrence is a useful heuristic for extracting specific protein-protein interaction from free texts.

There are two corresponding sets of rules in BioNLP specifying the patterns for identifying protein names and for extracting specific protein-protein interactions from free texts.

Identifying protein names can be challenging because the standard nomenclature is often only loosely followed by authors naming new proteins (D. Proux *et al.*, "Detecting gene symbols and names in biological texts: First step toward pertinent information extraction", Genome Informatics, 9:72--80, 1998; K. Fukuda *et al.*, "Toward information extraction: Identifying protein names from biological papers", Proc. Pacific Symposiùm on Biocomputing, 707--718, 1998). Even under the standard nomenclature, protein names can still be difficult to identify, as some of the protein names are long compound words or have multiple variants. To tackle this task, Fukuda *et al.* ("Toward information extraction: Identifying protein names from biological papers", Proc. Pacific Symposium on Biocomputing, 707--718, 1998) devised a set of rules to identify protein names based on lexical considerations such as the presence of upper cases and of special characters. Their lexical rules are incorporated in BioNLP and are augmented with the following strategies:

(i) Exclusion by standard dictionaries. BioNLP filters out most of the non-proper nouns in the abstracts by looking up the words in a classical dictionary.

(ii) Inclusion with semantic clues. Proper nouns that are not recognised, but are linked together by protein-protein interaction function words (e.g., "activate" or "inhibit"), are classified as potential new protein names.

(iii) Inclusion with protein dictionaries. BioNLP has a protein dictionary for the rapid identification of common protein names. This dictionary also allows the re-inclusion of protein names that are made up of the nouns excluded by (i). The dictionary may be manually edited by a user, or automatically learned from the protein-protein interactions subsequently extracted from the abstracts.

In addition, it is also possible to generalise BioNLP further to recognise names of small molecules and drugs. Then it would be possible to use BioNLP to extract interactions of proteins, small molecules, and drugs from scientific abstracts, as well as pure protein-protein interactions. The names of small molecules can be recognised by incorporating a dictionary of small molecules and lexical rules of the popular SMILES notations used for denoting names of small molecules. The names of drugs can be recognised by incorporating a dictionary of drug names, which can be obtained from various public and/or government drug registries.

Papers written to present specific results often contain information on subjects which are secondary to the main topic, but which may be quite useful for researchers working on new areas where a complete picture is unavailable and there are plenty of missing links to be filled. In a new field such as protein-protein interaction discovery, it is therefore important to extract as many protein-protein relationships as possible, including those that are only mentioned in the literature in a cursory manner, to build a reasonably complete interrelated network for the proteins of interest. The BioNLP rules are therefore designed to capture as many protein-protein relationships from the literature as possible, whether or not they are the main topic in the papers in which they are mentioned.

BioNLP maintains a set of function words for each interaction type. These function words can be edited by the user. Their roles are as keys into the literature for seeking out sentences that may contain protein-protein interaction information. For example, some of the key function words for the inhibit-activate relationship are

```
inhibitor: {inhibit, suppress, negatively regulate, … }
activator: {activate, induce, upregulate, positively regulate, …}
```

BioNLP seeks out sentences containing any of the function words and then searches for any protein names mentioned. These protein names are then associated with the function words using a suite of pattern matching rules to determine their actor-patient roles. Some examples of the pattern matching rules are shown below. In these examples, both `<A>` and `<B>` can denote individual or a conjunction of protein names, while `<fn>` denotes a matched function word:

(i) `<A>...<fn>...<B>`: This rule models the basic sentence pattern such as "A inhibits B, C, and D".

(ii) `<A>...<fn> of ...<B>`: This rules models sentences such as "A, an activator of B, is found to be lacking in the patient population".

(iii) `<A>...<fn> by ...<B>`: This models sentences in passive voice, such as "A is inhibited by the activities of B".

(iv) `<A>..., which ...<fn>...<B>,...`: This models sentences such as "A, which inhibits the activities of B, is found to be lacking in the patient population".

(v) `<fn> of <A> is ...<B>`: This template models sentences such as "Induction of A is caused by B".

The natural language processing described above is implemented in the BioNLP module as a program that processes the `articles` file retrieved by the BioKleisli-Abstracts module. It produces a file that we denote by the name `interactions` here. The file `interactions` has the following schema or type.

```
{ (#muid: num, #sentence: string, #matched: string,
   #interaction:  <#inhibit: (#actor: string, #patient: string),
                            #activate: (#actor: string, #patient: string)>) }
```

Hence the file is a set of records. Each record has the following fields. The field `muid` stores the unique identifier of the article in which a protein-protein interaction is extracted. The field `sentence` stores the sentence in the article in which the protein-protein interaction is extracted. The `matched` field stores the particular rule used to recognise that protein-protein interaction. The `interaction` field stores the protein-protein interaction extracted. The protein-protein interaction is stored either as an `inhibit` variant or as an `activate` variant. in either variant, the name of the protein in the `actor` role and the name of the protein in the `patient` role are stored.

In addition, if BioNLP is generalised to recognise also small molecules and drugs, it would also produce an additional file called `molecules`. The file `molecules` is a set of records. Each record stores the name of a molecule and its type (i.e., whether the molecule is a protein, a small molecule, or a drug).

THE LOGICAL REPRESENTATION

The files articles and interactions (and optionally molecules) constitute the logical representation of the extracted pathways. It can be thought of as a graph whose nodes are the actors and patients of the interaction extracted, whose directed edges connect up the interacting actors and patients, and these edges are annotated with the type of the interactions and associated information.

To make this concept more explicit, consider the following Kleisli program script that builds a "conceptual graph" from the interactions file.

```
writefile
  { (#edge-start: m.#actor, #edge-end: m.#patient, #edge-type: t,
     #edge-anno: { (
        #muid: u.#muid, #authors: p.#authors, #title: p.#title,
        #sentence: u.#sentence, #matched: u.#matched)
        | \u <- { i | \i ← interactions, i.#interaction = a},
          \p <- { i | \i ← articles, i.#muid = u.#muid })
   | \a ← set-unique { x.#interaction | \x ← interactions },
     (\t, \m) == case a
                 of <#inhibit: \m> → (" inhibit" , m)
                 or <#activate: \m> → (" activate" , m) }
to "conceptual-graph" using machineout;
```

This conceptual graph is a set of records or edges. Each record or edge has an edge-start field which stores the starting point or node of the edge, an edge-end field which stores the ending point or node of the edge, an edge-type field which stores whether the edge is "inhibit" or is "activate", and an edge-anno field which stores associated information of that edge. The associated information is the set of evidence from which the edge is derived. Each evidence comprises the sentence that mentions the interaction, the unique identifier (muid) of the article that contains that sentence, the authors and title of that article, and the BioNLP rule used to match that sentence matched).

Given the derivation of this conceptual graph, it is clear that we can manipulate it, and thus the extracted pathways, by manipulating the logical representation (namely the files articles and interactions). For example, to delete a node and edges involving this node in the graph, we can simply delete every interaction in the interactions file whose actor or patient is this node. We use this fact in the embodiment of the BioKleisli-Graphs module below.

In addition, suppose PIES re-executes the search keywords provided by the user and obtain more articles and more interactions. For the purpose of highlighting new discoveries, the logical representation is extended with two more files, delta-articles and delta-interactions. These two files have exactly the same schema as the files articles and interactions respectively. They respectively represent those articles and interactions that are strictly new. The embodiment of the BioKleisli-Graphs module provides a means for generating delta-articles and delta-interactions.

To make this concept more explicit, we can derive a "delta graph" from the conceptual graph described earlier and delta-articles and delta-interactions. The delta graph is just a copy of the conceptual graph, but each record is augmented with two additional fields: a is-new-interaction field which is set to true if and only if the interaction corresponding to that record is an interaction found in delta-interactions, and a has-new-evidence field which is set to true if and only if the edge-anno field of that record contains an muid of an article in delta-articles.

THE BIOKLEISLI-GRAPHS MODULE

The BioKleisli-Graphs module provides sophisticated manipulations on the extracted pathways by operating on the underlying logical representation. We describe each of these manipulations and their embodiment in the BioKleisli-Graphs module. We use the files `articles` and `interactions` to denote the logical representation of the current set of extracted pathways. We use the files `new-articles` and `new-interactions` to denote the resulting new logical representation of each manipulation. Note that the embodiments given here are chosen to maximise understanding rather than performance, as anyone skilled in the programming art would be able to produced more optimised (but harder to understand) implementations once the purpose of the embodiments is understood.

We first describe the "large-scale" manipulations that affect multiple proteins and interactions in the current set of pathways.

One of the manipulations is to allow the user (biologist) to specify a protein in the extracted pathways and then automatically delete that protein and its associated interactions from the extracted pathways. Let `KILL` denote the protein specified by the user. Then this manipulation corresponds to iterating through each `interaction` stored in the `interactions` file and keeping only those whose `actor` and `patient` is not the protein `KILL`, as implemented in the Kleisli program script below.

```
writefile
  { i
  | \i ← interactions,
    \x == case i.#interaction
           of <#inhibit: \y> ➔ y
           or <#activate: \y> ➔ y,
   not (x.#actor string-islike KILL),
   not (x.#patient string-islike KILL) }
to "new-interactions" using machineout;
```

One of the manipulations to allow the user (biologists) to specify a previous separately obtained set of pathways and to automatically merge it with the current extracted pathways. This manipulation is especially useful in a situation where the user would like to integrate a set of pathways he or some other biologists have previously saved into the current pathways he is investigating. Let a-articles and a-interactions denote the logical representation of the previous separately obtained pathways specified by the user. Then this manipulation is accomplished by taking the union of the files articles and a-articles and the union of the files interactions and a-interactions, eliminating all duplicated records. It is implemented in the Kleisli program script below, where {+} is the set union operator provided by the Kleisli query system.

```
writefile set-unique (articles {+} a-articles)
to "new-articles" using machineout;
writefile set-unique (interactions {+} a-interactions)
to "new-interactions" using machineout;
```

One of the manipulations is to allow the user (biologist) to provide some additional keywords, and then to automatically use these additional keywords to search the Internet for more scientific abstracts, and then to automatically extract more protein interaction information from these abstracts, and then to integrate these additional interaction information into the current set of pathways. This manipulation is especially useful in a situation where upon viewing the current set of pathways, the user makes some mental connections to some proteins he has hitherto ignored and would now like to investigate how these proteins might be connected to the current set of pathways he is studying. Let EXTRA-SPEC denote the set of additional keywords provided by the user. Then this manipulation is accomplished in several steps as described below:

- Retrieve the additional articles corresponding to EXTRA-SPEC using a program script similar to that of the BioKleisli-Abstracts module.
- Extract the interactions in these articles using BioNLP.
- Merge these additional articles and interactions using the merge operation we just described.

This operation is embodied in the Kleisli program script below.

```
writefile
  { x
  | \s ← EXTRA-SPEC,
    \u ← ml-get-uid-general (s),
    \x ←  ml-get-abstract-by-uid (u) }
to "a-articles" using machineout;
writefile
  {(#muid: x.#muid, #sentence: x.#sentence,
    #matched: x.#matched, #interaction: y)
  | \x ← process ("extract.pl",["a-articles"],"" ) using syscall-co,
    \y ← x.#interactions }
to "a-interactions" using machineout;
writefile set-unique (articles {+} a-articles)
to "new-articles" using machineout;
writefile set-unique (interactions {+} a-interactions)
to "new-interactions" using machineout;
```

One of the manipulations is to allow the user (biologist) to specify several proteins in the current set of pathways and then to automatically extract all interactions up-stream or down-stream of these proteins involving up to a specified number of intermediary proteins. This manipulation is particularly useful in the following two situations:

- the user wishes to export a specific portion of the current set of pathways that he is working on to another biologists.
- the user wishes to concentrate on a specific portion of the current set of pathways that he is working on. This manipulation is useful in this situations because as the set of pathways grows, it is likely that a set of proteins that are logically close together (that is, they interact through a small number of intermediary proteins) may be separated by a large distance in the graphical layout.

Let NODES denote the subset of proteins specified by the user. Let RADIUS denote the maximum number of intermediary proteins allowed by the user. Then this manipulation corresponds to the process of starting at each node in the conceptual graph that corresponds to a proteins in NODES and traversing the conceptual graph up to RADIUS many edges. In terms of the logical representation of the pathways, this manipulation can be accomplished by following these steps:

- Extract an undirected graph from the interactions file so that this graph represents which node is connected to which other node in the original logical representation.
- Compute a "transitive closure" of the undirected graph, but only up to RADIUS steps.
- Iterate over each interaction of the current interactions file and keep only those whose actor and patient are both in the "transitive closure" computed in the previous step.

A suggested implemented is given in the Kleisli program script below. Here fix is the fixpoint operator in Kleisli that satisfies the equation fix(F) = F(fix(F)) for any function F.

```
primitive undirected == set-unique
  { w
  | \i ← interactions,
    \a == case i.#interaction
          of <#inhibit: \z> → z
          or <#activate: \z> → z,
    \w ← { (#x: a.#actor, #y: a.#patient),
            (#x: a.#patient, #y: a.#actor) }};
primitive close ==
  let \F == fix (\f => (\r, \A) =>
    if r < 1
    then A
    else A {+} (f(r-1, { b.#y | \a ← A, \b ← undirected, b.#x = a })))
  in set-unique (F(RADIUS, NODES));
writefile
```

```
{ i
| \i ← interactions,
  \x == case i.#interaction
        of <#inhibit: \y> → y
        or <#activate: \y> → y,
  {x.#actor, x.#patient} set-subset close }
to "new-interactions" using machineout;
```

One of the manipulations is to allow a user to specify two differently-named proteins and to indicate that they are actually the same protein and then to automatically merge them and their associated interactions in the current set of pathways. This manipulation is useful because it is often the case that the same protein is named differently by different biologists. In terms of the conceptual graph, this manipulation implies that edges connected to the two nodes should now be connected to the merged node. Thus, it is equivalent to renaming the first of the two nodes to the second node. Let FIRST and SECOND denote the two specified proteins. Then, in terms of the logical representation of pathways, this manipulation can be accomplished by iterating through the file interactions and renaming each actor and patient that matches FIRST to SECOND. The Kleisli program script that implements this operation is given below:

```
primitive rename ==
  let \r1 == \x => if x string-islike FIRST then SECOND else x in
  let \r2 == \y => (#actor: y.#actor.r1, #patient: y.#patient.r1)
  in \i => case i
       of <#inhibit: \j> → <#inhibit: j.r2>
       or <#activate: \j> → <#activate: j.r2>;
writefile
  { (#muid: x.#muid, #sentence: x.#sentence,
     #matched: x.#matched, #interaction: x.#interaction.rename)
  | \x ← interactions }
to "new-interactions" using machineout;
```

One of the manipulations is to allow the user to specify a previously obtained set of pathways and then to automatically extract its differences from the current set of pathways. Recall that PIES regularly re-executes the search specified by the user. Thus, a use for this manipulation is for identifying what is new in the current set of pathways (obtained after a re-execution of the search) relative to an older set of pathways (obtained before the re-execution of the search). Let the files a-articles and a-interactions denote the logical representation of the specified previously obtained set of pathways. Then this manipulation is implemented in the Kleisli program script below by taking the set difference between the respective logical representations.

```
primitive delta-articles == articles set-diff a-articles;
primitive delta-interactions == interactions set-diff a-interactions;
```

As mentioned earlier, the `delta-articles` and `delta-interactions` can be used to highlight the new interactions found in the current set of pathways. When the pathways are displayed in a graphical visualisation, edges that correspond to `delta-interactions` (and thus the `is-new-interaction` field of the corresponding record in the delta graph is true) can be highlighted in red to indicate that they are new discoveries. Similarly, edges whose annotations reference some articles in `delta-articles` (and thus the `has-new-evidence` field of the corresponding record in the delta graph is true) can be highlighted in green to indicate that there are new evidence for these known interactions.

One of the manipulations is to allow the user to specify a protein and a new name and then to automatically introduce that new name into the current set of pathways as a new protein and to automatically replicate the interactions and other information of the specified protein for this new protein. In other words, this manipulation gives the newly named protein exactly the same interactions and information as the specified protein. Occasionally, two different groups of biologists may use the same name for two different proteins. Thus, if the abstracts of their papers were processed by the BioNLP module, the interactions of these two different proteins would be mapped to the same node. Let `NODE-A` denote this node. Then this manipulation can be used to rectify this problem as follows. The user uses this manipulation to replicate NODE-A and its interactions and other information to a new node, which we denote NODE-B here. Now the user can decide to let NODE-A be the protein of the first group of biologists and NODE-B be the protein of the second group of biologists. So he uses the manipulation for deleting individual interaction to delete appropriate interactions from NODE-A and NODE-B. The implementation of this manipulation in terms of the logical representation is an iteration through the file interactions and for each record that mentions NODE-A as the actor or the patient, a new copy of that record is made with NODE-A replaced by NODE-B.

We now describe the "small-scale" manipulations that affect one interaction at a time.

One of the manipulations is to allow the user to specify one interaction and to delete it from the current set of pathways. Let KILL denote the interaction to be deleted. Then it can be realised on the logical representation by an iteration through the file `interactions` and deleting each record whose `interaction` field matches KILL.

One of the manipulations is to allow the user to specify one interaction and to reverse its direction. Let REVERSE denote the specified interaction. It can be implementation in terms of the logical representation by an iteration through the file `interactions` and modify each record whose `interaction` field matches REVERSE by interchanging its `actor` and `patient` fields. The following Kleisli program script is a possible implementation.

```
writefile
    {(#muid: x.#muid, #sentence: x.#sentence, #matched: x.#matched,
      #interaction: i)
    | \x ← interactions,
      x.#interaction = REVERSE,
      \i == case REVERSE
             of <#inhibit:\y> → (#actor:y.#patient, #patient:y.#actor)
```

```
            or <#activate:\y> → (#actor:y.#patient, #patient:y.#actor)}
to " new-interactions" using machineout;
```

One of the manipulations is to allow the user to specify one interaction and to invert its nature by changing it from an inhibition to an activation or by changing it from an activation to an inhibition. Let INVERT be the specified interaction. It can be implemented in terms of the logical representation by an iteration through the interactions file and modify each record whose interaction field matches INVERT by changing that field from an inhibit variant to an activate variant. The following Kleisli program script is a possible implementation.

```
writefile
    {(#muid: x.#muid, #sentence: x.#sentence, #matched: x.#matched,
      #interaction: i)
    | \x ← interactions,
      x.#interaction = INVERT,
      \i == case INVERT
            of <#inhibit:\y> → <#activate: y>
            or <#activate:\y> → <#inhibit: y>}
to " new-interactions" using machineout;
```

One of the manipulations is to allow the user to specify a new interaction and to insert it into the current set of pathways. In terms of the logical representation, this manipulation is a straightforward addition of a new record into the interactions file.

THE GRAPH LAYOUT MODULE

The "graph" corresponding to the logical representation is a conceptual one. It has nodes and edges that connect these nodes. To make it a physical one that is visible in a two dimensional screen, we need to assign to each node a x-y co-ordinate that indicates its position on the screen, so that directed lines or arcs connecting the nodes can be drawn to represent the edges. The Graph Layout module is a program for computing suitable x-y co-ordinates to assign to the nodes so that when displayed, the graph will be visually pleasant.

The Graph Layout module is based on an algorithm for drawing directed graphs previously disclosed in the paper, ER Gansner *et al.*, "A Technique for Drawing Directed Graphs", IEEE Trans. Software Engineering, 19(3):214--230, 1993. We describe the basic idea here. For convenience of description, we layout the graph in a top-to-bottom manner on the screen. The method can be easily adapted for a left-to-right layout.

First the graph is divided into connected components and each connected component is layout separately. Given a connected component, a depth-first traversal is used to obtain a tree from it. Each node in the tree is assigned a rank. The node at the root of the tree is assigned rank 0, its children rank 1, and so on. The screen is divided into horizontal strips. Nodes at rank k is assigned to strip k. Each strip is divided into as many vertical regions as there are nodes assigned to this trip. For each strip, the nodes assigned to it are sorted to minimise crossing of connecting arcs. So a node of rank k and is in position j in the sorted ordering among all nodes at that rank is assigned to region j of strip k. The initial co-ordinates for that node is the centre of that region.

THE BIOJAKE MODULE

The BioJAKE module is based on the BioJAKE system previously disclosed in Salamonsen *et al.*, "BioJAKE: A tool for the creation, visualisation, and manipulation of metabolic pathways", Proc. Pacific Symposium on Biocomputing, 392--400, 1999. The original system is modified to suit the additional functionalities of PIES. We describe these modifications below.

It accepts as input a logical representation of a set of pathways (the files `articles`, `interactions`, and optionally `molecules`, `delta-articles`, and `delta-interactions`). It then constructs the conceptual (delta) graph corresponding to this logical representation. Then it uses the Graph Layout module to compute the preliminary x-y co-ordinates for the nodes in this graph. The nodes are then displayed according to these preliminary x-y co-ordinates. Arrowed lines are then drawn to represent edges connecting the nodes according to the graph. An arrowed line is labelled with a plus sign (+) if the edge-type field corresponding to that edge is "inhibit" and is labelled with a minus sign (-) otherwise. If the is-new-interaction field of that edge is true, red colour is used for that line, as a means to indicate that this line is a new interaction. If the is-new-interaction field is false and the has-new-evidence field is true, green colour is used for that line, as a means to indicate that this line is an existing interaction for which there is a new evidence. Otherwise, black colour is used for the line. If the molecules file is provided, the nodes are displayed using different graphical icons so that nodes denoting proteins, small molecules, and drugs can be differentiated visually.

After displaying the graph as above, the user is allowed to freely modify the display by the means of drag-and-drop and point-and-click to re-position the nodes and connecting arrowed lines as desired. He is also allowed to carry out other BioJAKE operations disclosed in Salamonsen *et al.*, "BioJAKE: A tool for the creation, visualisation, and manipulation of metabolic pathways", Proc. Pacific Symposium on Biocomputing, 392--400, 1999.

A menu is also added to BioJAKE to allow the user to invoke the additional ("large-scale") manipulation operations provided by the BioKleisli-Graphs module. Note that the "small-scale" manipulations of the BioKleisli-Graphs module are already available in the original implementation of BioJAKE disclosed in Salamonsen *et al.*, "BioJAKE: A tool for the creation, visualisation, and manipulation of metabolic pathways", *Proc. Pacific Symposium on Biocomputing*, 392—400, 1999.

WE CLAIM:

1. A system for the routine automatic discovery and presentation of biological pathways from online text abstracts including:

input means for inputting search criteria;

first processing means for scanning information from online sources for a match of the search criteria;

first storage means for storing abstracts located matching the search criteria;

second processing means for analysing the stored abstracts and extracting precise protein interaction information from them;

second storage means for storing the extracted information; and

display means for graphically representing the extracted protein interaction information.

2. A system as claimed in claim 1, further including:

editing means for editing the graphical representation and / or enabling queries and/or links to the World Wide Web to be attached to elements of the graphical representation.

3. A system as claimed in claim 1 or 2, further including:

updating means for updating the graphical representation at predetermined intervals by performing the following tasks:

a. invoking means to retrieve and store new abstracts satisfying the given specification;

b. invoking means to extract and store new protein interaction information from them; and

c. invoking means to alert the user.

4. A system as claimed in claim 1, in which the display means includes:

highlighting means adapted to highlight the graphical elements that correspond to the newly discovered protein interaction information;

editing means adapted to enable the graphical presentation to be edited and to enable queries and/or links to the World Wide Web to be attached to the graphical elements.

5. A system as claimed in claim 1, in which the extracting means is adapted to extract other biological information in addition to the protein interaction information.

6. A method of providing biological pathways from on-line text abstracts, the method including

a. scanning information related to a selected topic;

b. extracting information based on predetermined criteria;

c. generating an initial graphical representation of the extracted information, wherein the improvement comprises:

d. providing an updated graphical representation by reiterating a, b and c on new information, not yet scanned; and

e. alerting a user to the presence of new extracted information based on the new information.

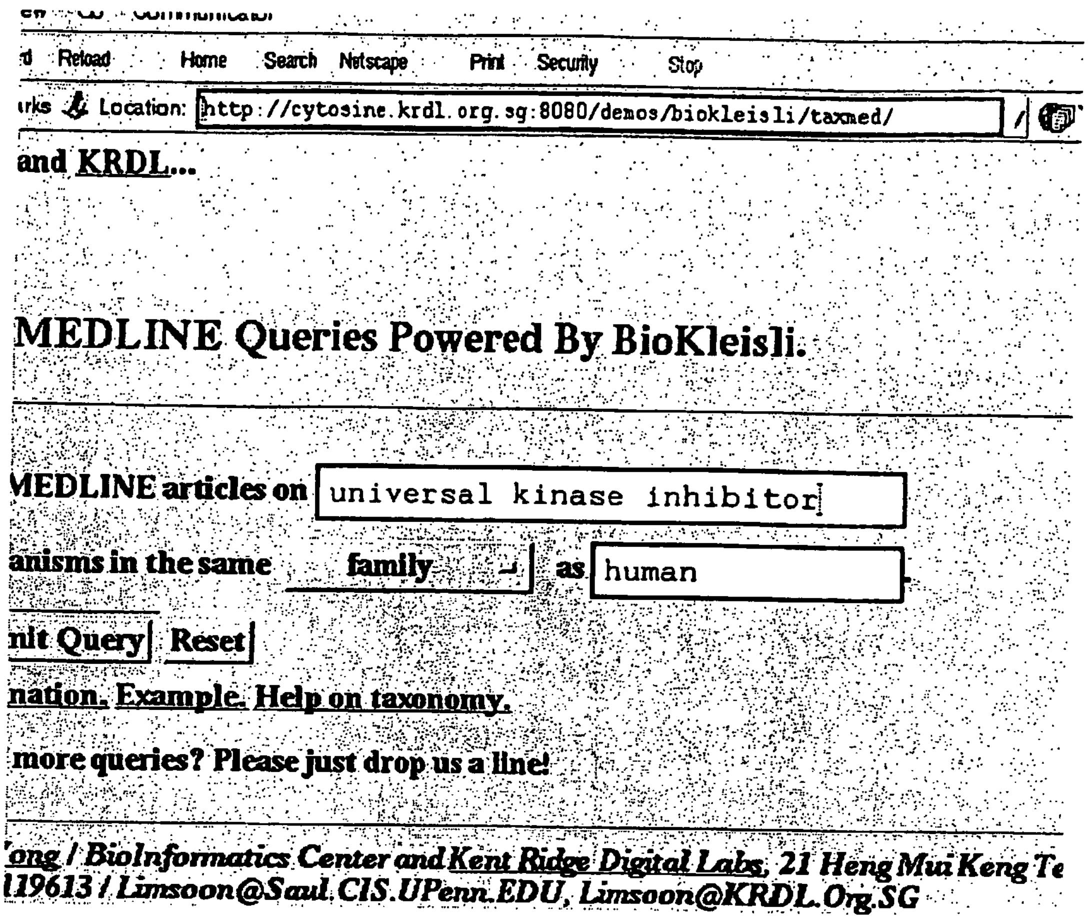
7. A method as claimed in claim 6, in which the extracted information is obtained using natural language processing.

8. A method as claimed in claim 7, in which the processing is that as disclosed in SK Ng, M. Wong, "Toward automatic pathway discovery from on-line scientific text abstracts", *Genome Informatics*, 10:104—112, December 1999.

9. A method as claimed in claim 6, in which the extracted information is obtained using a keyword based search.

10. A method as claimed in any one of claims 6 to 9, in which the updated representation is represented in a visually distinctive manner compared to the initial graphical representation.

11. A method as claimed in claim 10, in which the visually distinctive manner is to show a different colour for each update.



http://cytosine.krdl.org.sg:8080/demos/biokleisli/taxmed/

and KRDL...

MEDLINE Queries Powered By BioKleisli.

MEDLINE articles on universal kinase inhibitor

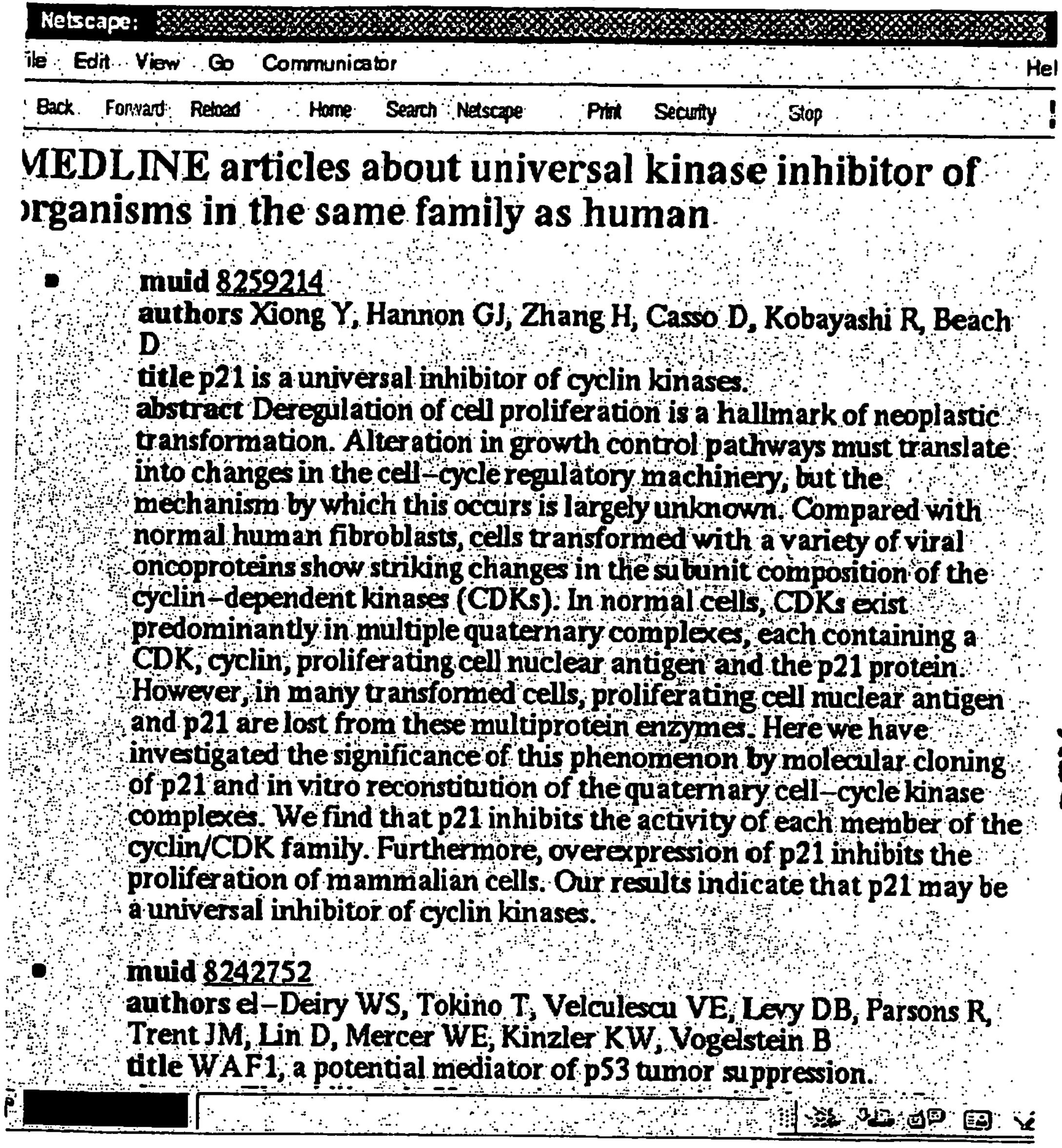
anisms in the same family as human

nation. Example. Help on taxonomy.

more queries? Please just drop us a line!

ong / BioInformatics Center and Kent Ridge Digital Labs, 21 Heng Mui Keng Te
119613 / Limsoon@Saul.CIS.UPenn.EDU, Limsoon@KRDL.Org.SG

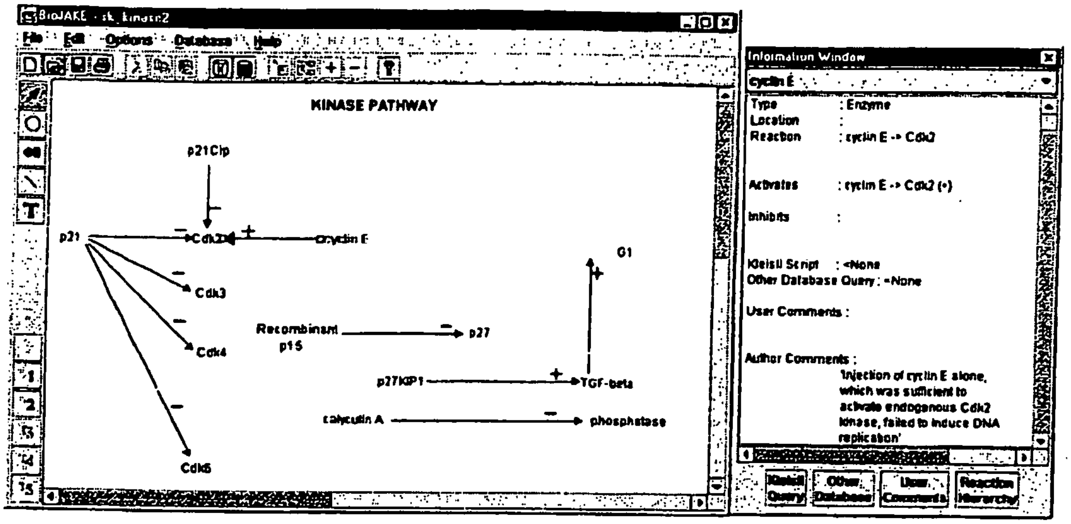
FIGURE 1


Netscape:
ile Edit View Go Communicator Hel
Back Forward Reload Home Search Netscape Print Security Stop

MEDLINE articles about universal kinase inhibitor of organisms in the same family as human

- **muid** 8259214
  authors Xiong Y, Hannon GJ, Zhang H, Casso D, Kobayashi R, Beach D
  title p21 is a universal inhibitor of cyclin kinases.
  abstract Deregulation of cell proliferation is a hallmark of neoplastic transformation. Alteration in growth control pathways must translate into changes in the cell-cycle regulatory machinery, but the mechanism by which this occurs is largely unknown. Compared with normal human fibroblasts, cells transformed with a variety of viral oncoproteins show striking changes in the subunit composition of the cyclin-dependent kinases (CDKs). In normal cells, CDKs exist predominantly in multiple quaternary complexes, each containing a CDK, cyclin, proliferating cell nuclear antigen and the p21 protein. However, in many transformed cells, proliferating cell nuclear antigen and p21 are lost from these multiprotein enzymes. Here we have investigated the significance of this phenomenon by molecular cloning of p21 and in vitro reconstitution of the quaternary cell-cycle kinase complexes. We find that p21 inhibits the activity of each member of the cyclin/CDK family. Furthermore, overexpression of p21 inhibits the proliferation of mammalian cells. Our results indicate that p21 may be a universal inhibitor of cyclin kinases.
- **muid** 8242752
  authors el-Deiry WS, Tokino T, Velculescu VE, Levy DB, Parsons R, Trent JM, Lin D, Mercer WE, Kinzler KW, Vogelstein B
  title WAF1, a potential mediator of p53 tumor suppression.

FIGURE 2



Type : Enzyme
Location :
Reaction : cyclin E -> Cdk2

Activates : cyclin E -> Cdk2 (+)

Inhibits :

Kleisli Script : <None
Other Database Query : <None

User Comments :

Author Comments :
'Injection of cyclin E alone, which was sufficient to activate endogenous Cdk2 kinase, failed to induce DNA replication'

FIGURE 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SG 01/00217

CLASSIFICATION OF SUBJECT MATTER

IPC^7: G06F 17/30, G06N 5/02

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC^7: G06F, G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

XPESP, WPI, PAJ, EPODOC, ScienceDirect

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 00/49540 A (Cellmocis inc.) 24 August 2000 (24.08.00) *claims 1-36.*	1-11
A	S.M. Paley and P. D. Karp "Adapting EcoCyc for use on World Wide Web",in: Gene Vol. 172 (1), pages 43-50, 1 June 1996 (01.06.96) *the whole document.*	1-11
A	P.D. Karp "Metabolic Databases, in: Trends in Biochemical Sciences, Vol. 23 (2), pages 114-116, 1 March 1998 (01.03.98) *the whole document.*	1-11

□ Further documents are listed in the continuation of Box C.   ☒ See patent family annex.

* Special categories of cited documents:

„A" document defining the general state of the art which is not considered to be of particular relevance

„E" earlier application or patent but published on or after the international filing date

„L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

„O" document referring to an oral disclosure, use, exhibition or other means

„P" document published prior to the international filing date but later than the priority date claimed

„T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

„X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

„Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

„&" document member of the same patent family

Date of the actual completion of the international search 28 December 2001 (28.12.2001)	Date of mailing of the international search report 28 February 2002 (28.02.2002)
Name and mailing adress of the ISA/AT Austrian Patent Office Kohlmarkt 8-10; A-1014 Vienna Facsimile No. 1/53424/535	Authorized officer WERNER Telephone No. 1/53424/357

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SG 01/00217

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☒ Claims Nos.: 8
because they relate to subject matter not required to be searched by this Authority, namely:
instead of defining certain characterizing features, a scientific paper is cited.

2. □ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. □ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. □ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. □ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. □ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. □ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

□ The additional search fees were accompanied by the applicant's protest.
□ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/SG 01/00217

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO A 0049540		none	